

# Analisa Kinerja Penggunaan Kode (LDPC) Low Density Parity Check Code Pada Kanal Multipath Fading

Hamka, Yoedi Moegiharto<sup>2</sup>

<sup>1</sup>Mahasiswa Politeknik Elektronika Negeri Surabaya, Jurusan Teknik Telekomunikasi

<sup>2</sup>Dosen Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember  
Kampus ITS, Surabaya 60111

e-mail : [hamtarolucu@yahoo.com](mailto:hamtarolucu@yahoo.com) e-mail : [ymoegiharto@eepis-its.edu](mailto:ymoegiharto@eepis-its.edu)

## Abstrak

Tujuan dari pembuatan proyek akhir ini adalah menganalisa kinerja *Low Density Parity Check Code* (LDPC) pada kanal AWGN dan *Multipath fading* berbasis perangkat lunak.

*Encoding* yang dilakukan pada sisi transmitter dengan menggunakan metode *sparse matriks* berbasis *Approximation Lower Triangular* dengan *Coderate*  $\frac{1}{2}$ . Sedangkan pada sisi penerima dilakukan dengan metode *decoding* menggunakan *decoding sum product log domain*.

Dengan hasilnya adalah berupa nilai BER (*bit error rate*) yaitu membandingkan kinerja *Low Density Parity Check Code* melalui kanal AWGN dan saat melewati *Multipath fading* dengan metode *decoding Log Domain Sum Product*.

Pada perbedaan ukuran matriks, hasil yang dicapai tidak terlalu ada perbedaan dalam kinerja LDPC. Kemudian pada perbedaan nilai iterasi, semakin banyak iterasi, mempengaruhi kinerja LDPC sehingga hasil yang dicapai semakin baik. Target BER  $10^{-3}$  pada kanal AWGN tercapai pada rata-rata nilai 3.9 dB. Sedangkan pada kanal *multipath fading* pada nilai 6.57 dB.

Sehingga kinerja *Low Density Parity Check Code* melalui kanal AWGN lebih bagus dibandingkan saat melewati *Multipath fading* dengan metode *decoding Log Domain Sum Product*

Kata Kunci : *Low Density Parity Check Code*, *sparse matriks* berbasis *Approximation Lower Triangular*, Kanal *Multipath Fading*, *decoding sum product log domain*, BER (*Bit Error Rate*)

## 1. Pendahuluan

Pada zaman sekarang semakin banyak kebutuhan yang diperlukan dalam dunia teknologi. Baik dari bentuk komunikasi, system yang handal, integrasi *wireless* dengan berbagai layanan yang bermacam-macam. Pada proses pengiriman maupun penerimaan data semakin besar ukuran data yang dibutuhkan, yang mana semakin besar pula resiko kesalahan dalam transmisi data tersebut. Sehingga diperlukan

proses dalam penyeleksian sehingga data yang dikirim dapat sampai pada penerima dengan baik. Apalagi saat seperti ini banyak terdapat gangguan atau *noise* yang ditimbulkan oleh lingkungan sekitar terhadap terjalannya sebuah komunikasi.

Sehingga di perlukan suatu proses untuk mengoreksi dan mendeteksi atau disebut *error control coding* agar data atau informasi pada saat dikirim maupun di terima dapat menghasilkan dengan baik. Banyak metode yang di gunakan untuk mengoreksi dan mendeteksi kesalahan yang terjadi. Terdapat pula macam-macam karakteristik kelebihan dan kekurangan dari setiap metode tersebut.

Salah satu metode tersebut adalah *Low Density Parity Check (LDPC) Code*. *Low Density Parity Check Code* adalah salah satu kelas dari linear block kode yang memiliki kepadatan rendah, nama itu berasal dari karakteristik *parity check*-nya yang hanya berisi sedikit bit "1" jika dibandingkan dengan jumlah bit "0". Keuntungan dari penggunaan LDPC adalah dapat menyediakan performa yang sangat mendekati nilai kapasitas dari berbagai macam kanal dan mempunyai proses *decoding* yang linier dan diharapkan dengan pengiriman *high-bit-rate* dapat membantu untuk menghasilkan probabilitas kesalahan bit yang rendah.

*Low density Parity Check Code* juga dapat digunakan untuk pengiriman dan penerimaan data yang berukuran besar. Dengan metode ini akan di implementasikan dengan melewati sebuah kanal *multipath fading*. Untuk dapat mengetahui bagaimana pengaruh yang didapat ketika sebuah pendeteksian dan pengkoreksian kesalahan dilewatkan sebuah kanal.

## 2. Teori Penunjang Latar Belakang LDPC

LDPC pertama kali ditemukan oleh *Galagher* pada 1960 dan hampir tidak dianggap

lagi sekitar 35 tahun kemudian. Kemudian muncul Tanner tahun 1982 yaitu mengembangkan LDPC code dan memperkenalkan representasi grafik LDPC code, atau dikenal dengan *Tanner Graph*. Pembelajaran tentang LDPC code dihidupkan kembali sekitar 1990, yang dikerjakan oleh MacKay, Luby, dan lain-lain. Ketika pertama kali ditemukan, karena batas dari daya computer saat itu yang belum tinggi, kode ini dianggap tidak praktis, dan terlupakan. Saat ini dengan kehebatan daya computer yang telah hadir, kode LDPC telah digunakan kembali dan termasuk kedalam kode dengan performa yang terbaik pada situasi *low density*. Ternyata dengan sistem *Gallagher* merupakan keuntungan dari *linear block code* yang menggunakan parity check matrik dengan *sparse* (*low density* atau jumlah elemen *non-zero* yang sedikit).

## Representasi Code LDPC

### Representasi matriks

LDPC adalah *linear block code* dimana *parity check matriks*  $H$  mempunyai kerapatan rendah yang mempunyai nilai bit 1, *regular* LDPC adalah *linear block code* yang memiliki *parity check matriks*  $H$  yang berisi  $\omega_c$  jumlah bit 1 pada tiap kolom dan  $\omega_r = \omega_c(n/m)$  adalah jumlah bit 1 pada tiap baris. Dimana  $\omega_c \ll m$  (*equivalent*,  $\omega_c \ll m$ ). Kode rate  $R = k/n$  dihubungkan dengan parameter melalui  $R = 1 - \omega_c/\omega_r$  (diasumsikan  $H$  adalah full rank). Jika  $H$  adalah *sparse* atau kerapatannya rendah, tapi jumlah dari bit 1 pada tiap baris atau kolom tidak konstan disebut dengan *irregular* LDPC.

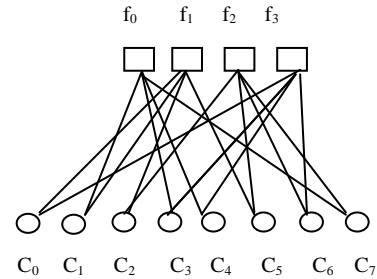
$$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

**Gambar 2.1** representasi matriks

### Representasi grafik

Tanner memperkenalkan sebuah representasi grafik untuk code-code LDPC. Grafik *Tanner* adalah *bipartite graph* yaitu suatu grafik yang yang tidak langsung berhubungan tetapi memisahkan antara dua kelas, dimana *edge* hanya terhubung dengan dua *node* tetapi tidak bertempat pada kelas yang sama. Ada dua

type *nodes* dalam *Graph Tanner*, yaitu *variable nodes* (*v-nodes*) dan *check nodes* (*c-nodes*), dan yang menyambungkan antara *v-node* dan *c-node* disebut *edge*.



**Gambar 2.2** representasi grafik

Pada gambar 2.2 grafik tanner menghubungkan  $H$ . Dengan mengamati  $c_1, c_3, c_4$  dan  $c_7$  dihubungkan dengan *c-node* yang disebut  $f_0$  pada  $H$ ,  $h_{01}, h_{03}, h_{04}, h_{07} = 1$  (sedangkan yang lain adalah nol). Kemudian langkah selanjutnya yaitu *c-node*  $f_1, f_2, f_3$ , mengikuti baris 1, 2, 3 pada  $H$ . Dengan catatan dengan mengikuti persamaan  $cH^T = 0$ . Nilai bit yang dihubungkan pada *c-node* yang sama harus berjumlah 0. Kita dapat memproses terus kolom untuk membangun grafik *tanner*

### Regular dan irregular code LDPC

Code LDPC disebut dengan *regular* jika  $\omega_c$  adalah bernilai konstan Pada untuk setiap kolom dan  $\omega_r$  pada setiap barisnya. Pada contoh adalah *regular* : tiap *v node* memiliki dua koneksi *edge* dan tiap *c-node* memiliki 4 koneksi *edge* (pada tiap *v-node* adalah 2 dan tiap *c-node* ada 4). Jika  $H$  adalah *low density* tetapi jumlah 1 pada tiap baris ataupun kolom tidak konstan dinamakan code *irregular* LDPC.

Pada code LDPC terdapat :

$\omega_c$  adalah jumlah bit 1 pada tiap kolom  
 $\omega_r = \omega_c (n/m)$  adalah jumlah bit 1 pada tiap baris  
 $R = 1 - \omega_c/\omega_r$  adalah kode rate

### Konstruksi code LDPC

Ada banyak cara untuk menkonstruksi code LDPC antara lain :

1. Code *Gallagher*
2. Code *Mackay*

## Decoding LDPC

1. *Bit flipping (BF) decoding*
2. *Algoritma decoding log domain Sum product*

### Proses Decoding LDPC

#### 1. *Hard Decision decoding (Bit-flip)*

Ilustrasi proses *decoding iterative* menggunakan algoritma *bit flip*, didasarkan pada penandaan *hard decision* (0 atau 1) untuk setiap bit yang diterima. Bagian yang penting dari *decoding iterative* adalah pelewatan pesan (*message passing*) antar *node* pada grafik *tanner*. Pada algoritma *bit flip* pesannya sederhana: sebuah *message node* mengirimkan pesan pada setiap *check node* yang terhubung mendeklarasikan dirinya 0 atau 1. Dan setiap *check node* mengirimkan pesan ke setiap *message node* yang terhubung pada grafik *tanner*, apakah cek paritas terpenuhi atau tidak, algoritma *sum product* LDPC memiliki prosedur yang sama, hanya saja pesan yang di pertukarkan lebih kompleks.

Algoritma *decoding bit flip* :

- Langkah 1: inisialisasi: setiap *bit node* menandai nilai bit yang diterima kanal dan mengirimkan pesan ke *check node* yang terhubung pada grafik *tanner* mengindikasikan nilainya.
- Langkah 2: *Parity update*: dengan menggunakan pesan dari *bit node*, setiap *check node* mengecek apakah persamaan check paritas terpenuhi jika seluruh persamaan cek paritas terpenuhi algoritma berhenti, Jika tidak setiap *check node* mengirimkan pesan ke *bit node* yang terhubung mengindikasikan apakah persamaan cek paritas terpenuhi atau tidak.
- Langkah 3 : *Bit update* ; jika mayoritas pesan yang diterima setiap *message node* tidak memenuhi, *message node* merubah (*flip*) nilai saat ini. Kemudian kembali ke langkah 2, jika jumlah iterasi maksimum terlampaui dan *codeword* belum *valid*, maka algoritma berhenti dan pesan *failure to converge* dilaporkan.

Proses pendekodean LDPC dapat menggunakan algoritma *probability propagation* yang disebut juga dengan algoritma *sum product* atau *belief propagation* yang diimplementasikan menggunakan *factor graph* yang terdiri dua buah node yaitu *bit node* dan *check node*. Setiap *check node* berhubungan dengan kolom dari *parity check matriks* yang juga merupakan *codeword*. Setiap *check node* berhubungan dengan baris pada matriks *parity checks*

#### 2. *Decoding Log Domain Sum Product*

1. Proses dari *check node* menuju *bit node*

Setiap *check node*  $m$  mengumpulkan seluruh informasi yang masuk,  $L(q_{l \rightarrow m})$ 's dan memperbaharui kekuatan bit 1 berdasarkan pada informasi seluruh bit yang dihubungkan pada *check node*  $m$

$$L(r_{m \rightarrow l}) = 2 \tanh^{-1} \prod_{l' \in L(m) \setminus l} \tanh\left(\frac{L(q_{l' \rightarrow m})}{2}\right)$$

2. Proses dari *Bit node* menuju *check node*

$$L(q_{l \rightarrow m}) = L(p_l) + \sum_{m' \in E(l) \setminus m} L(r_{m' \rightarrow l})$$

3. Akhir proses *decoder*

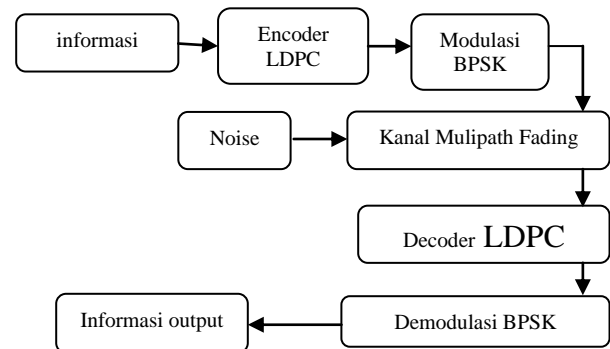
Proses *decoder* menghasilkan peluang *a posteriori* untuk menghasilkan menjumlahkan informasi dari seluruh *check nodes* yang menghubungkan bit  $l$

$$L(q_l) = L(p_l) + \sum_{m \in E(l)} L(r_{m' \rightarrow l})$$

### 3. Metodologi

Untuk membuat proyek akhir ini (Analisa Kinerja *Low Density Parity Check Code* pada Kanal *Multipath Fading*) maka dibutuhkan langkah-langkah sebagai berikut :

Perancangan, dan Pembuatan Sistem



Gambar 3.1 Blok Diagram LDPC

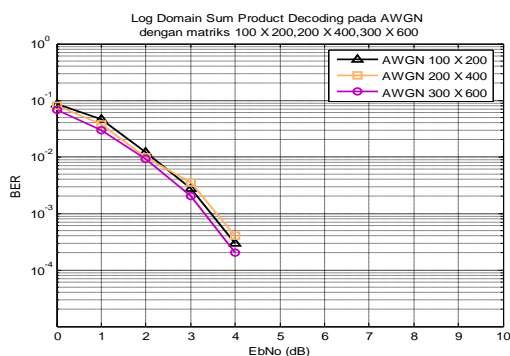
Dari diagram blok diatas dapat dilihat bahwa informasi akan di masukkan ke *encoder* LDPC, kemudian *output encoder* akan dimodulasi dengan menggunakan modulasi BPSK yang kemudian dikirim melalui kanal *Multipath fading*, disini kanal *Fading* akan dtambahkan dengan *noise*, keluaran kanal *Multipath fading* bercampur *noise* akan diterima oleh *decoder* di sisi terima yang sebelumnya di demodulasi terlebih dahulu, dari *output decoder* LDPC akan dihasilkan *output*. Bahasa pemrograman yang digunakan pada perancangan system ini berupa *software matlab*, jadi semua blok pada perancangan system ini berupa *software under matlab*.

Informasi yang dikirimkan berupa data biner untuk *encoder* LDPC data informasi yang berupa data biner dikodekan kembali menjadi bit-bit *codeword*, *codeword* inilah yang kemudian akan dikirim atau diproses melalu kanal *Multipat fading* dengan ditambah *noise*, *codeword* ditambah *noise* awgn tersebut akan diterima oleh *decoder* LDPC, disini data *codeword* bercampur *noise* tadi dikodekan kembali dan menghasilkan *output*, *output* ini akan berupa bit-bit informasi.

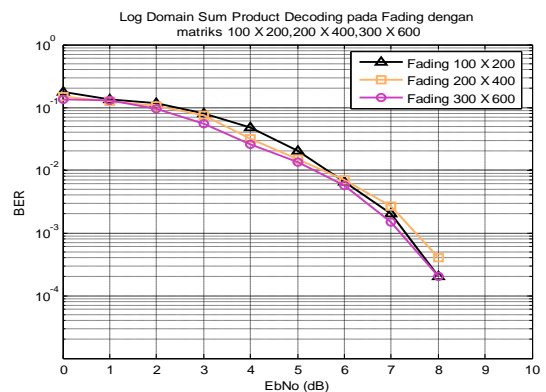
#### 4. Pengujian dan analisa

##### Pengujian

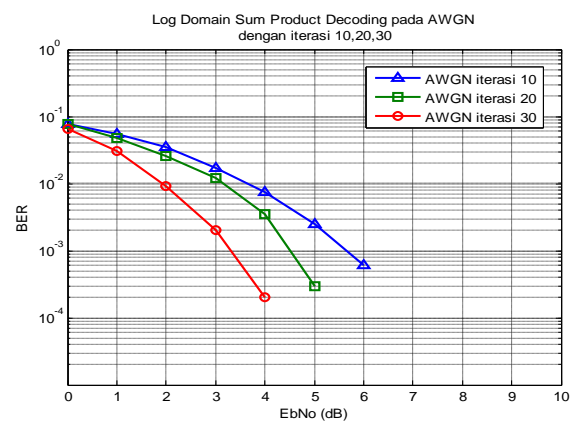
Setelah pembuatan sistem selesai dilakukan, maka sistem akan di uji dan di analisa. Dari pengujian tesebut didapatkan data-data sebagai berikut :



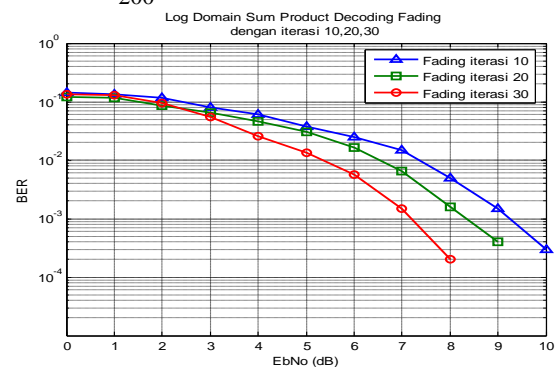
**Gambar 4.1** Hasil perbandingan tiga ukuran matrik pada kanal AWGN dengan iterasi 30



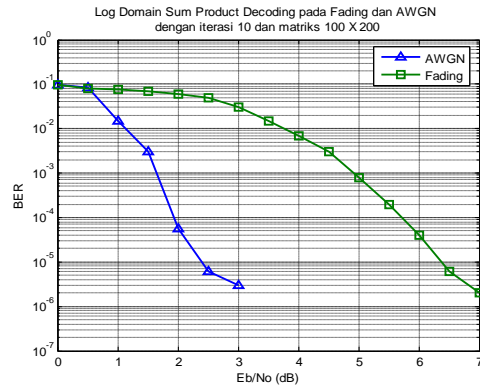
**Gambar 4.2** Hasil perbandingan tiga ukuran matriks pada kanal *Fading* Dengan iterasi 30



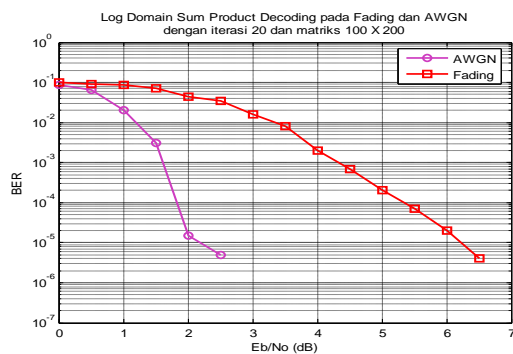
**Gambar 4.3** Hasil perbandingan tiga ukuran iterasi pada kanal AWGN dengan ukuran matriks 100 X 200



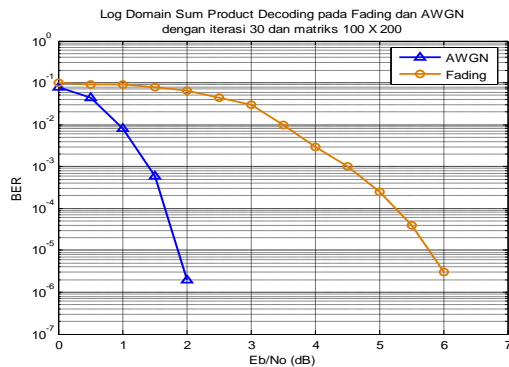
**Gambar 4.4** Hasil perbandingan tiga ukuran iterasi pada kanal *Rayleigh Fading* dengan matriks 100 X 200



**Gambar 4.5** Hasil perbandingan AWGN dan *Fading* dengan Iterasi 10 dan matriks 100 X 200.



**Gambar 4.6** Hasil perbandingan AWGN dan *Fading* dengan Iterasi 20 dan matriks 100 X 200.



**Gambar 4.7** Hasil perbandingan AWGN dan *Fading* dengan Iterasi 30 dan matriks 100 X 200.

### Analisa

Hasil simulasi dapat dilihat bahwa seiring dengan perubahan ukuran matriks yang semakin besar maka performa sistem dari teknik pengkodean LDPC mengalami perubahan yang tidak terlalu besar. Hal ini dapat dilihat pada sistem LDPC dengan ukuran matriks 100 X 200 pada kanal AWGN dengan target BER  $10^{-3}$

dapat dicapai pada  $E_b/N_0 = 3.30$  dB. Sedangkan untuk ukuran matriks 200 X 400 target BER tercapai pada  $E_b/N_0 = 3.46$  dB, pada ukuran matriks 300 X 600 target BER tercapai pada  $E_b/N_0 = 3.57$  dB.

Sedangkan pada system LDPC dengan ukuran matriks 100 X 200 pada kanal *Rayleigh fading* dengan target BER  $10^{-3}$  dapat dicapai pada  $E_b/N_0 = 7.2$  dB. Sedangkan untuk ukuran matriks 200 X 400 target BER tercapai pada  $E_b/N_0 = 7.3$  dB, pada ukuran matriks 300 X 600 target BER tercapai pada  $E_b/N_0 = 7.5$  dB.

Kemudian pengamatan perbandingan nilai iterasi yang digunakan. Pada BER  $10^{-3}$  dengan iterasi 10 di  $E_b/N_0 = 3.35$  dB, pada iterasi 20 di  $E_b/N_0 = 4.42$  dB, pada iterasi 30 di  $E_b/N_0 = 5.64$  dB

Pada kanal *Rayleigh fading*, pengamatan pada BER  $10^{-3}$  dengan iterasi 10 di  $E_b/N_0 = 7.2$  dB, pada iterasi 20 di  $E_b/N_0 = 8.33$  dB, pada iterasi 30 di  $E_b/N_0 = 9$  dB.

Jika diperhatikan pada setiap kenaikan jumlah iterasi *decoding*, maka dapat dilihat terjadi perbaikan performansi berupa *coding gain*. Hasil yang didapatkan berdasarkan grafik tersebut yaitu nilai optimal untuk iterasi *decoding* adalah 30. Karena untuk iterasi 10 dan 20 terjadi peningkatan *coding gain* yang tidak terlalu besar. Meskipun dari hal ketiga iterasi yang digunakan terjadi peningkatan *coding gain* sebesar 1 dB. Mengingat bahwa jumlah iterasi mempengaruhi proses *delay decoding*, maka jumlah iterasi yang optimal adalah 30 untuk jumlah iterasi *decoding log domain sum product* LDPC.

### 5. KESIMPULAN

Berdasarkan pada hasil simulasi dan analisa yang telah dibahas pada bab 4, dapat diketahui kinerja dari LDPC yang dihasilkan adalah dalam bentuk ukuran matriks saat melewati kanal AWGN dan *Rayleigh Fading*, perbandingan nilai iterasi pada kanal AWGN dan *Rayleigh fading*. Setelah semua tahap dilakukan, yaitu penyusunan bab 1 sampai bab 4 maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Perbedaan ukuran matriks tidak terlalu mempengaruhi hasil *error* yang dihasilkan karena  $\omega_c$  dan  $\omega_r$  telah ditentukan nilainya.

2. Jumlah iterasi mempengaruhi performansi sistem. Dimana terjadi *coding gain* sebesar  $\pm 2$ dB untuk kanal AWGN dan  $\pm 6$  dB untuk *Rayleigh fading* untuk iterasi yang terbaik.

## 6.SARAN

Program simulasi kinerja kode LDPC pada kanal *multipath fading* yang telah selesai masih terdapat kekurangan sehingga masih harus diperbaiki dan dikembangkan untuk mendapatkan hasil yang maksimal pada penelitian selanjutnya sehingga perlu dilakukan perbaikan dalam hal berikut :

1. Bit informasi yang dikirimkan lebih banyak lagi tetapi proses yang dibutuhkan untuk mendapatkan probabilitas kesalahan bit akan membutuhkan waktu yang cukup lama
2. *Coderate* LDPC harus berbeda-beda agar hasil yang terbaik pada *Code rate* LDPC dapat diterapkan pada kanal *multipath fading*

## 7. Daftar Pustaka :

- [1] Sarah J.Johnson, Steven R. Weller, Low Density parity-check codes: Design and decoding. School of Electrical Engineering and Computer Science University of Newcastle. 2002
- [2] G.Gallager, Low-Density Parity-Check Codes, Cambridge, MA:M.I.T, Press. 1963
- [3] Radford M Neal. Sparse Matrix Methods and Probabilistic Interference Algoritma,Dept. of Statistics and Dept. of Computer Science University of Toronto. 1999
- [4] Feras A.K Al-Zuraiqi, “ Analysis, Simulation and Modelling Of Mobile and Fixed Fading Channel”, June. 2004
- [5] Sun Jian,” An Introduction to Low Density Parity Check (LDPC) Codes”, June 2003
- [6] Bernhard M.J. Leiner, LDPC Codes– a brief Tutorial, April. 2005
- [7] William E. Ryan, “ An Introduction to LDPC codes”, August. 2003
- [8] Marius Pop, “Statistical Analysis of Sum-of-Sinusoids Fading Channel Simulators”, February, 1999.
- [9] R. H. Clarke, "A Statistical Theory of Mobile-Radio Reception", *Bell Syst. Tech. J.*, pp. 957- 1000, Jul.-Aug. 1968.
- [10] L. W. Couch, “Digital and Analog Communication Systems”, Prentice Hall, 1997